Answers to Assignment 2: Due Thursday February 13, 2020

1. Write a regular expression for the language consisting of all strings over \{a, b\} which contain the substring \textit{aaa}.

\[(a + b)^*aaa(a + b)^*\]

2. Use the method given on page 86 of the sixth edition of Linz, or on page 89 of the fifth edition, to find a regular expression equivalent to the following NFA.

In our construction, to avoid clutter, we do not write any self-loop labeled \(\lambda\), and we do not write any arc labeled \(\emptyset\). This omissions is justified by the identities \(\lambda x = x \lambda = x\), \(\emptyset x = x \emptyset = \emptyset\), where \(x\) is any regular expression.

Eliminate \(\textcircled{1}\):

Eliminate \(\textcircled{2}\):

The regular expression is \((a + b)^*a(a + b)(a + b)\).

3. The following DFA accepts the language consisting of all binary numerals for positive multiples of three, where a leading 0 is allowed. Use the method given on page 86 of the sixth edition of Linz, or on page 89 of the fifth edition, to find an equivalent regular expression.
Since the start state is final, introduce a dummy start state:

```
0  1
λ  0
3  1  0
```

Eliminate ①:

```
0  1
λ  1
3  0
```

Eliminate ②:

```
0+101*01
```

The regular expression is \((0 + 101^*01)^*\).

4. (a) State the pumping lemma for regular languages.

For any regular language \(L\) there exists a positive integer \(p\), the pumping length of \(L\), such that for any \(w \in L\) whose length is at least \(p\) there exist strings \(x, y, z\) such that:

i. \(w = xyz\),
ii. \(|y| \geq 1\),
iii. \(|xy| \leq p\),
iv. for any integer \(i > 0\), \(xy^i z \in L\).

(b) Use the pumping lemma to prove that the language \(L = \{a^n b^n : n \geq 0\}\) is not regular.

Assume \(L\) is regular. Let \(p\) be a pumping length of \(L\). Let \(w = a^p b^p\). Then there exist strings \(x, y, z\) such that

i. \(w = xyz\),
ii. \(|y| \geq 1\),
iii. \(|xy| \leq p\),
iv. for any integer \(i > 0\), \(xy^i z \in L\).

By (ii) \(|y| = k\) for some \(k > 0\). By (i) and (iii) \(xy\) contains no \(b\), hence \(y\) contains to \(b\). By (iv) \(xy^0 z = a^{p-k} b^p \in L\), contradiction because \(p - k \neq p\). Thus \(L\) is not regular.
5. Work problem 9(a) on page 138 of the sixth edition, which is problem 7(a) on page 137 of the fifth edition.

\[
S \rightarrow AAAT \\
A \rightarrow a | \lambda \\
T \rightarrow aTb | Tb | \lambda
\]

6. Work problem 9(c) on page 138 of the sixth edition, which is problem 7(c) on page 137 of the fifth edition.

\[
S \rightarrow S_1 | S_2 \\
S_1 \rightarrow aS_1 | aT_1 \\
T_1 \rightarrow aaT_1b | \lambda \\
S_2 \rightarrow S_2b | T_1b \\
T_2 \rightarrow aaT_2b | \lambda
\]


The following unambiguous grammar generates the language. There are also ambiguous grammars for the same language.

\[
S \rightarrow (S)S | [S]S | \lambda
\]


Let \( R \) be the language of all regular expressions over \( \{a, b\} \).

The alphabet for \( R \) is \( \Sigma = \{a, b, \lambda, 0, +, *, (, )\} \) There is a simple ambiguous context-free grammar for \( R \):

\[
S \rightarrow SS | S + S | (S) | S^* | (S) | a | b | \lambda | \emptyset
\]

On the other hand, we might prefer an unambiguous context-free grammar. We use the usual method of constructing a grammar for an algebraic language, using variables \( E, T, \) and \( F, \) where \( E \) is the start symbol.

\[
E \rightarrow E + T | T \\
T \rightarrow TF | F \\
F \rightarrow F^* | (E) | a | b | \lambda | \emptyset
\]